Please amend the claims to read as indicated in the following list of claims:

1. [Currently amended] A method of enabling second party to prove to a third party the existence of ~~to verify~~ an association between the second party and a first party, the first party being associated with a first element[[,]] of a first algebraic group, ~~and a~~ the second party being associated with a second element, of a second algebraic group, formed from an identifier string of the second party using a hash function, and there being ~~wherein: there exists~~ a computable bilinear map for the first and second elements; wherein a second-party computer entity, acting on behalf of the second party: ~~the first party has a first secret and computes a first product from the first secret and the first element; the second party has both a second secret, and~~

receives a shared secret provided by the first party as the product of ~~the~~ a first secret and the second element;

~~the second party~~ computes first, second and third verification parameters as the product of ~~the~~ a second secret with said shared secret, the second element and the first element respectively; and

outputs the first, second and third verification parameters for use by the third party in proving the association between the first and second parties.

2. [Currently amended] A method according to claim 1, wherein the second-party computer entity generates a further shared secret from the second secret and an

identifier string of a fourth party, the second party outputting ~~passing~~ this further shared secret to the fourth party for use by the latter as the private key of a public/private key pair the public key of which is formed by the identifier string of the fourth party.

3. [Original] A method according to claim 1, wherein the first and second parties are respectively parent and child trusted authorities in a hierarchy of trusted authorities.

4. [Original] A method according to claim 1, wherein the first and second algebraic groups are the same.

5. [Original] A method according to claim 1, wherein the first and second elements are points on the same elliptic curve.

6. [Currently amended] A method of verifying an association between the first and second parties of claim 1 by using a function p providing said bilinear map; the method comprising a third-party computer entity carrying out the following operations using the ~~non-secret data elements~~ verification parameters of claim 1:

computing the second element from the identifier string of the second party;

carrying out a first check:

p(third verification parameter, computed second element) = p(first element, second verification parameter)

carrying ~~carries~~ out a second check:

p(first element, first verification parameter) = p(first product, second verification parameter) where said

first product is a public parameter provided by the first party and corresponds to the product of the first secret and the first element;

the association between the first and second parties being treated as verified if both checks are passed.

7. [Original] A method according to claim 6, wherein said bilinear mapping function is based on a Tate or Weil pairing.

8. [Currently amended] A method of verifying an association between a first party associated with a first element, of a first algebraic group, and a second party associated with a second element, of a second algebraic group [[;]], the first and second elements being such that there exists a bilinear mapping p for these elements[[;]], the method comprising a third-party computer entity carrying out the following operations:

receiving both data indicative of said first element, and a first product formed by the first party from a first secret and the first element;

receiving in respect of the second party both an identifier string, and first, second and third verification parameters;

computing the second element from the identifier string of the second party;

carrying out a first check:

p(third verification parameter, computed second element)= p(first element, second verification parameter)

carrying out a second check:

p(first element, first verification parameter)=
p(first product, second verification parameter)
the association between the first and second parties being treated as verified if both checks are passed.

9. [Original] A method according to claim 8, wherein said bilinear mapping function is based on a Tate or Weil pairing.

10. [Original] A method according to claim 8, wherein the first and second algebraic groups are the same.

11. [Original] A method according to claim 8, wherein the first and second elements are points on the same elliptic curve.

12. [Currently amended] A method of enabling verification of an association between parties, the method comprising:

generating a first private key and public key for a first party;

generating a second private and public key for a second party wherein the second private key is derived from the first private key and second public key; ~~and~~

generating a third private key for the second party that is used in association with the first public key, the second private key and the second public key to form a first cryptographic parameter, a second cryptographic parameter and a third public key respectively; and

outputting the first, second, and third cryptographic parameters.

13. [Original] A method according to claim 12, wherein a third party uses the first, second and third cryptographic parameters together with the first and second public keys to check, by using bilinear mapping, whether there is an association between the first and second parties.

14. [Original] A method according to claim 12, wherein the bilinear mapping is based on either a Tate or Weil pairing.

15. [Original] A method according to claim 12, wherein the third private key is combined with a third party's public key to form an associated private key such that an association can be established between the third public key of the second party and the first public key of the first party.

16. [Original] A method according to claim 12, wherein the third private key is a random number.

17. [Original] A method according to claim 12, wherein the first party is a first trusted party and the second party is a second trusted party.

Claim 18. Cancelled.

19. [Currently amended] Apparatus arranged to enable a third party to verify an association between the apparatus and a first party that has a first secret and is associated with a first element of a first algebraic group, the apparatus being associated with a second element, of a second algebraic group, and the first and second elements

being such that there exists a bilinear mapping *p* for these elements; the apparatus comprising:

a memory for holding a second secret and an identifier string associated with the apparatus,

means for forming said second element from said identifier string using a hash function,

means for receiving from the first party a shared secret based on said first secret and said first element, and for storing this shared secret in the memory,

means for computing first, second and third verification parameters as the product of the second secret with said shared secret, said second element and said first element respectively, and

means for making available said identifier string and said verification parameters to the third party.

20. [Original] Apparatus according to claim 19, wherein the first and second algebraic groups are the same.

21. [Original] A method according to claim 19, wherein the first and second elements are points on the same elliptic curve.

22. [Currently amended] Apparatus for verifying an association between a first party associated with a first element, of a first algebraic group, and a second party associated with a second element, of a second algebraic group; the first and second elements being such that there exists a bilinear mapping *p* for these elements; the apparatus comprising:

means for receiving both data indicative of the first element, and a first product formed by the first party from a first secret and the first element[[:]];

means for receiving in respect of the second party both an identifier string, and first, second and third verification parameters;

means for computing the second element from the identifier string of the second party using a hash function;

means for carrying out a first check:

p(third verification parameter, computed second element) = p(first element, second verification parameter);

means for carrying out a second check:

p(first element, first verification parameter) = p(first product, second verification parameter);

means responsive to both checks being passed, to confirm that there exists an association between the first and second parties.

23. [Original] Apparatus according to claim 22, wherein said bilinear mapping p is based on a Tate or Weil pairing.

24. [Original] Apparatus according to claim 22, wherein the first and second elements are points on the same elliptic curve.

25. [Currently amended] An hierarchy of trusted authorities wherein:

each trusted authority is associated with a point on an elliptic curve, this point being derived, at least for

each non-root trusted authority, from an identifier string of the trusted authority using a hash function;

at least the non-leaf trusted authorities each has a standard elliptic-curve public/private key pair wherein the private key is formed by a secret of the trusted authority concerned and the public key comprises the product of this secret with the point associated with that trusted authority;

at least the non-root trusted authorities each has an identifier-based elliptic-curve public/private key pair wherein the public key comprises the identifier string of the trusted authority concerned and the private key is a shared secret provided by a said trusted authority at a next level up in the hierarchy, the shared secret being the product of the secret of the next-level-up trusted authority and the point associated with the trusted authority to which the shared secret is provided; and

at least the non-root trusted authorities each has two further public parameters formed by the product of the secret of the trusted authority respectively with the shared secret provided to it by the next-level-up trusted authority and with the point associated with the latter.

Claim 26. Cancelled.

27. [Currently amended] A computer program product stored on computer readable media for use in generating verification parameters to enable a third party to verify an association between a first party that has a first secret and is associated with a first element, of a first algebraic group, and computing apparatus associated with a

second element, of a second algebraic group; the first and second elements being such that there exists a bilinear mapping *p* for these elements; the program product being arranged, when installed in said computing apparatus, to condition the apparatus for:

storing, in a memory of the apparatus, a second secret and an identifier string associated with the apparatus,

forming the second element from said identifier string using a hash function,

receiving from the first party a shared secret based on said first secret and said first element, and for storing this shared secret in said memory, ~~and~~

computing first, second and third verification parameters as the product of the second secret with said shared secret, said second element and said first element respectively, and

outputting the first, second and third verification parameters for use by the third party.

28. [Currently amended] A computer program product stored on computer readable media for use in verifying an association between a first party associated with a first element, of a first algebraic group, and a second party associated with a second element, of a second algebraic group[[;]], the first and second elements being such that there exists a bilinear mapping *p* for these elements [[;]], the program product being arranged, when installed in computing apparatus, to condition the apparatus for:

receiving both data indicative of the first element, and a first product formed by the first party from a first secret and the first element[[:]];

receiving in respect of the second party both an identifier string, and first, second and third verification parameters;

computing the second element from the identifier string of the second party using a hash function;

carrying out a first check:

p(third verification parameter, computed second element) = p(first element, second verification parameter);

carrying out a second check:

p(first element, first verification parameter) = p(first product, second verification parameter);

confirming the existence of an association between the first and second parties means if both checks are passed.